Maps

HashMap

TreeMap

LinkeHashMap

Map un container in care putem sa asociem o cheie cu o valoare. Principiul de dictionar.

Entry<K,V> inner class pt a stoca variabile.

Map-ul isi face resize-ing automat.

Implementarile de la Seturi folosesc Map-uri.

ConccurentHashMap

* E threadsafe

Inserarea si cautarea e in timp constant.

Elementele nu apar in ordine, ci intr-un mod nedeterminist din cauza hashului.

Cele 3 nu sunt threadSafe

HashMap – fast lookup

LinkedList – LRU Least Recently Used o colectie de dim fixa ce retine elementele pe baza accesarii acestora.Care a fost cel mai putin fol e inlocuit, e folositor pt cache-uri.

- le retine in ordinea inserarii

-mai rapid cand iteram

TreeMap – red black tree, sortate in functie de comporatorul dat de noi

-metode: put, get, entrySet().iterator, keySet,values, containsKey, containsValue, remove

-se poate adauga o cheie nula

equals()

mostenita de toate obiectele din Java

compara doua obiecte

default compara referintele

returneaza un boolean

reflexive, simetrica, consistenta , ob non-null ob.equals(null) mereu fals

hashCode()

folosita cand se face inserarea si cautarea unui element

returneaza un int, care e hashcode-ul

e calculate fol o fct matematica

In Java dimensiunea default pt bucket e 16 ( sa fie puteri ale lui 2)

Daca hashcode ul depaseste valoarea se face resize

Daca avem doua obiecte egale, au acelasi hashCode

Daca au acelasi hashCode nu trebuie sa fie egale

String are deja implementat hashCode